10. ročník Žilina 2. – 4. 7. 2018
Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach
osscconf.soit.sk
OSS Conf

OSSConf 2026: 1–7

Raylib + R = Raylib × R = RaylibR: Ahoj, světe!

PAVEL STŘÍŽ (CZ)

Abstrakt. Článek stručně odkazuje na instalaci Raylibu (prostředí pro 2D a 3D grafiku napsaný primárně v programovacím jazyce C), ale též na mnohem důležitější RaylibR. Jedná se o most mezi Raylibem a výpočetním prostředí R. Užívá k tomu eRkový balíček `Rcpp`. Byl to do určité míry boj, který však stál za to.

Klíčová slova. Raylib, R, RaylibR, Rcpp.

Raylib + R = Raylib × R = RaylibR: HELLO, WORLD!

Abstract. The article briefly refers to an installation of Raylib (an environment for 2D and 3D graphics written primarily in the C programming language), but most importantly to an installation of RaylibR. That's a wrapper of the Raylib environment for the R environment written with the help of the `Rcpp` package. It was a bit of struggle, but it paid its dividends in the end.

Keywords. Raylib, R, RaylibR, Rcpp.

1. Jak jsem do toho spadl

Musím se přiznat, že nejsem Céčkař (nedejbůh Cépépéčkař, zatím) ani eRkař, aspoň se tak necítím, učím se za pochodu. Během programování sudoku s překryvy (angl. multi-sudoku puzzle; multi-grid sudoku; overlapping sudoku) jsem měl nápad udělat si 2D a 3D vizualizaci ze zadání do řešení. Ve stylu arkádovek 80. let, či aspoň nějak podobně. TeX umí hodně věcí, ale tohle není jeho doména, resp. má doména v TeXu, tak jsem hledal alternativu.

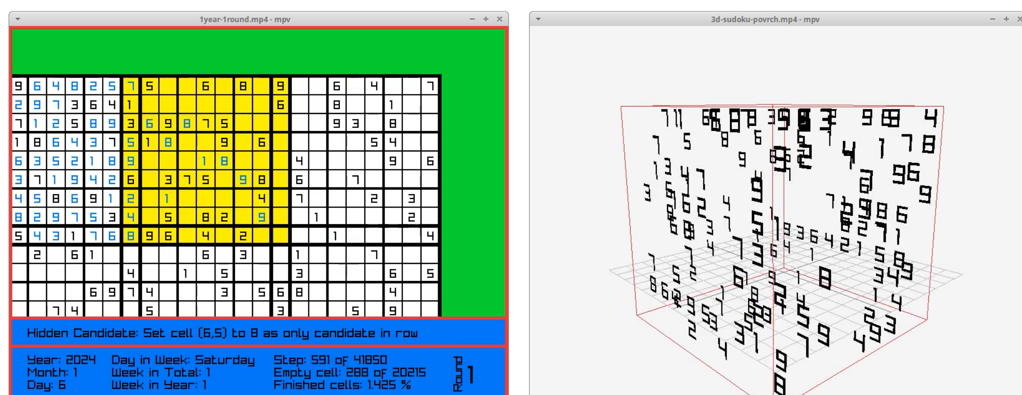
Rudolf Blaško má hezké výstupy přes Asymptote, ale na ten jsem rezignoval, potřeboval jsem ještě větší flexibilitu (myšleno programátorskou svobodu). Na vlastní prostředí jsem to také neviděl.

Zkusil jsem tedy řadu existujících grafických prostředí pro C, moc se mi líbí (od X11, FLTK, ImGui, GTK+ přes Qt, Cairo, SFML až po SDL2), objevil jsem i malé projekty typu Olive.c a v tom čase jsem narazil na Raylib od Ramona Santamarie. Užívá GLFW a OpenGL. Zrovna na YouTube slavil 10 let existence projektu. Objev byl učiněn hlavně díky tomuto videu `https://www.youtube.com/watch?v=0To1aYglVHE`, kde uživatel Tsoding Daily zkoumá uložení projektu do videa.

Prošel jsem webové stránky a zjistil, že by to mělo běžet na většině operačních systémů, platforem a ve všech známějších programovacích jazycích. Ani jsem

nepomyslel na Python, hned jsem skočil po nativní verzi v C. Za pomoci seznamu příkazů, `https://www.raylib.com/cheatsheet/cheatsheet.html`, a několika videí na YouTube, to začalo běžet.

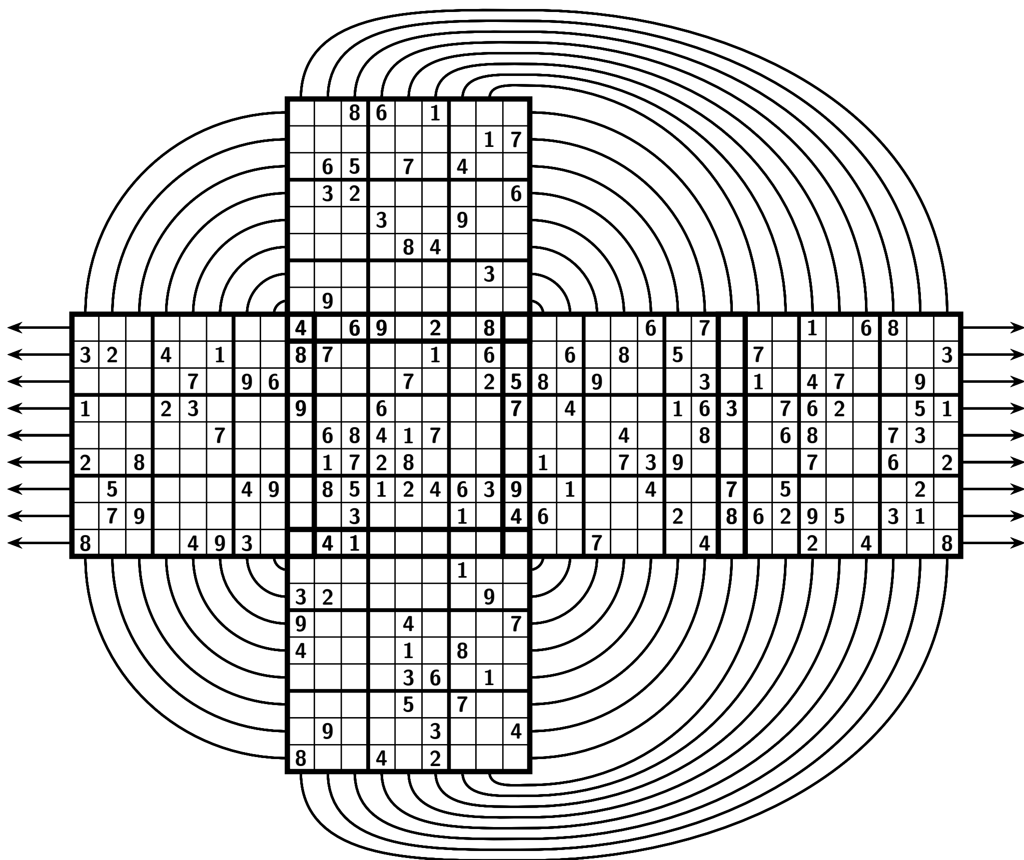
Přikládám dvě ukázky. Není tam běhající či létající panáček či panenka, bylo potřeba se držet svých mantinelů. Sudoku generuje Lua, to jsou mé experimenty do Vánoce 2023, a Python3, to jsou nové pokusy od povánočních svátků dál. 2D rozkres krychle zadání a řešení níž.



2. Vzorek z vlastní zahrádky

Na této straně je řešení, na další straně zadání 3D sudoku (povrch krychle, tedy 6 sudoku), rozkresleného na ploše se vztahy. Vztahy jsou komplikovanější (hrany sdílí dvě sudoku, rohy dokonce tři), ale princip vzniku je podobný.

								7	4	8	6	9	1	3	5	2																
								9	2	3	8	4	5	6	1	7																
								1	6	5	2	7	3	4	9	8																
								5	3	2	7	1	9	8	4	6																
								8	1	4	3	2	6	9	7	5																
								6	7	9	5	8	4	1	2	3																
								2	8	1	4	6	7	5	3	9																
								3	9	7	1	5	8	2	6	4																
7	9	1	5	8	6	2	3	4	5	6	9	3	2	7	8	1	4	9	3	5	6	8	7	2	5	3	1	9	6	8	4	7
3	2	6	4	9	1	5	7	8	7	2	5	4	1	9	6	3	7	6	4	8	2	5	1	9	7	4	5	8	2	1	6	3
5	8	4	3	7	2	9	6	1	3	9	8	7	6	4	2	5	8	2	9	1	7	4	3	6	1	8	4	7	3	2	9	5
1	6	7	2	3	5	8	4	9	2	4	6	5	3	8	1	7	5	4	2	9	8	1	6	3	8	7	6	2	9	4	5	1
9	4	5	8	1	7	6	2	3	6	8	4	1	7	5	9	2	9	3	6	4	1	7	8	5	2	6	8	4	1	7	3	9
2	3	8	9	6	4	7	1	5	1	7	2	8	9	3	4	6	1	8	5	7	3	9	2	4	9	1	7	3	5	6	8	2
6	5	3	1	2	8	4	9	7	8	5	1	2	4	6	3	9	2	1	8	6	4	3	5	7	4	5	3	1	8	9	2	6
4	7	9	6	5	3	1	8	2	9	3	7	6	8	1	5	4	6	7	1	3	5	2	9	8	6	2	9	5	7	3	1	4
8	1	2	7	4	9	3	5	6	4	1	3	9	5	2	7	8	3	5	7	2	9	6	4	1	3	9	2	6	4	5	7	8
								5	8	9	6	2	7	1	4	3																
								3	2	7	1	8	4	6	9	5																
								9	1	2	5	4	8	3	6	7																
								4	6	3	7	1	9	8	5	2																
								7	5	8	2	3	6	4	1	9																
								2	3	4	9	5	1	7	8	6																
								1	9	6	8	7	3	5	2	4																
								8	7	5	4	6	2	9	3	1																



3. Instalace knihovny Raylib

Co byl pro mne problém byla instalace, jak pracuji záměrně na starém notebooku, protože co mi běží na něm, poběží i na rychlejších strojích, toť základní idea. Hlavní zádrhel byl, že nemám OpenGL verzi 3.3, ale nižší (v2.1), tedy se to při instalaci knihovny a ukázek musí nastavovat. Svou verzi zjistíte na Linuxu přes knihovnu `mesa-utils`:

```
$ glxinfo | grep "OpenGL version"
```

V principu jsem následoval návod, přes `sudo make install` mám knihovny pro Desktop verzi, lokálně pak pro Web verzi (tu používám minimálně). Tím nenápadně naznačuji, že Raylib umí generovat své výstupy pro webové prohlížeče – za pomocí WebAssembly.

Když mé náhledy videoukázek viděl Aleš Kozubík vznesl dotaz, jestli by nebylo možné Raylib aktivovat v R, že by se jim tam něco takového hodilo. Na první dvě kola jsem R v seznamu jazyků na Raylibu na mobilu přehlédl, užil jsem RaylibRS,

jak je v logu písmeno R, to bylo však pro Rust. Pak jsem si očistil brýle a na normálním monitoru to našel! Ve vyhledávači jsem se přímo zeptal na RaylibR.

4. Instalace RaylibR

Instalovat Raylib sólově netřeba, jen si stačí pobrat závislé balíčky.

```
sudo apt install build-essential git cmake libasound2-dev
  libx11-dev libxrandr-dev libxi-dev libgl1-mesa-dev
  libglu1-mesa-dev libxcursor-dev libxinerama-dev
```

Zkusil jsem v R:

```
> install.packages("Rcpp")
> install.packages("remotes")
> remotes::install_github("jeroenjanssens/raylibr")
```

Ale instalace mi zkolabovala, neb nemůže najít `R_ext/Error.h`. Začal mi trochu boj. Podezříval jsem `Rcpp`, ale nebylo to tím. Na GitHubu v *Open Issues* je tento problém jako jediný otevřený (k datu 17. 2. 2024), ale tipy na řešení mi nezabraly, tak jsem zkoumal. Našel jsem dvě cesty.

První, byť druhá v pořadí vzniku, je tato. Soubor mi v počítači existuje, jen v trochu jiné složce. Jestli je to dané tím, že jsem R instaloval z linuxového repozitáře versus přímo verzi ze CRANu, nevím. V `~/.bashrc` jsem si přidal:

```
export C_INCLUDE_PATH=$C_INCLUDE_PATH:/usr/share/R/include
```

Je to vlastně snadné, když člověk ví, co hledat. Má první cesta byla o něco bojovnější. Stáhl jsem si z GitHubu RaylibR. Rozbalil jsem `raylibr-main.zip`. Nic s termínem `Error.h` jsem nenašel. Po hlubším zkoumání jsem zjistil, že Raylib je přítomen ve složce `inst/` jako `raylib-4.0.0-modified. tar.gz`, rozbalil jsem jej, a pak už to bylo „snadné“. Ve složce `src/` je `Makefile`, do kterého jsem zasáhl. V mém případě na řádku 207 odkomentováním:

```
GRAPHICS = GRAPHICS_API_OPENGL_21
```

Ale co je důležitější, přidal jsem na řádku 379 na konec parametr na mou cestu s žádaným souborem, do tvaru:

```
INCLUDE_PATHS = -I. -Iexternal/glfw/include -Iexternal/glfw/deps/mingw
  -I$(R_HOME)/include -I/usr/share/R/include
```

Zabalil jsem složku `raylib-4.0.0-modified` do `tar.gz`. A chybí už jen jeden krok, jít o dvě úrovně výš a zabalit složku `raylibr-main` do souboru `raylibr-main.tar.gz`. Pozor, R (zatím) neumí pracovat se `zip` soubory.

V R už to pak bylo přímočaré:

```
> install.packages("raylibr-main.tar.gz",repo=NULL)
```

Radost to byla veliká, byť takto psané mi to přijde vše jasné, stručné a logické. Své dva objevy jsem připsal autorovi na GitHub, je velká šance, že i tato chyba bude uzavřena.

5. Několik ukázek

Seznam ukázek získáme v R přes:

```
> library(raylibr)
> demo(package="raylibr")
```

Zkusit si jednu z nich, *Hello, World!*, můžeme takto:

```
> demo("helloworld",package="raylibr") # nebo
> demo(raylibr::helloworld) # či dokonce jen demo(helloworld)
```

Alternativa zobrazení dostupných ukázek v R je:

```
> help.start()
  Vybrat: Packages --> raylibr --> Code demos.
```

Ukončení R je příkazem `q()` nebo `quit()`. Volíme ano (y) či ne (n), chceme-li si uložit pracovní prostředí, nechceme-li práci zatím přerušit volíme pokračovat (c).

V krátké přednášce na YouTube Jeroen Janssens, autor RaylibR, zmínil své dva vzorky – `raylibr::stroop` v úvodu povídání a `raylibr::beatbox` v jejím závěru.

A mj. ty ukázky jsou interaktivní, takže toho hada si můžete zahrát, v tom bludišti skutečně můžete chodit ap. Více ukázek (přes sto) hledejte přímo na stránkách Raylibu, v RaylibR jich je „jen“ jedenáct. Po instalaci jsem `*.R` našel v ~`/R/x86_64-pc-linux-gnu-library/4.3/raylibr/demo/`.

Pro úplnost článku uvádím i zdrojový kód pro R.

```
library(raylibr)
init_window(600,400,"R & Raylib: Hello, World!")
while (!window_should_close()) {
  alpha<-abs(sin(get_time()))
  begin_drawing()
  clear_background("black")
  draw_circle(300,200,seq(150,10,by=-10),c("red","white"))
  draw_text(c("Hello,","World!"),225,c(120,220),64,fade("black",alpha))
  draw_fps(10,10)
  end_drawing()
  }
close_window()
```

Soubor se jmenuje `helloworld.R` a spustíme si jej z příkazového řádku přes:

```
$ Rscript helloworld.R
```

Okno s Raylibem se zavře přes klávesu Escape.

6. Závěrem

Asi by byla pro mne zajímavá výzva zkusit své „arkádové“ pokusy (Raylib+C) překlopit do Raylib+R a podat o zkušenostech s konverzí zprávu, ale nejdu do toho. Jsou další úkoly. Možná by šel udělat automat na konverzi, nechávám jako otevřený problém.

Musím se přiznat, že to byl pro mne boj, i Raylib i poté RaylibR, ale zvítězil jsem. Těším se, až nahodím silnější stroj s OpenGL v3.3 a zkusím si instalace ještě jednou. O tom možná pár slov na konferenci OSSConf, do termínu uzávěrky sborníku testy skoro určitě nestihnu. Hlavně se těším na ukázky v Raylibu (speciálně složka `shaders/`), ne všechny mi plně jely. Možná to bude tím, že pro v2.0 a v2.1 není podpora ve složce `examples/shaders/ resources/shaders/`, ale jen pro starší OpenGL v1.0, v1.2, a pak až pro novější v3.3…

Držím s instalacemi palce, nenechte se odradit, pokud vám něco na prvních deset pokusů nejede!

Zde jsou mé oblíbené ukázky: `raylibr::balls` za 2D a `raylibr::cubes` za 3D grafiku, už běžící přes RaylibR.



7. Post Scriptum

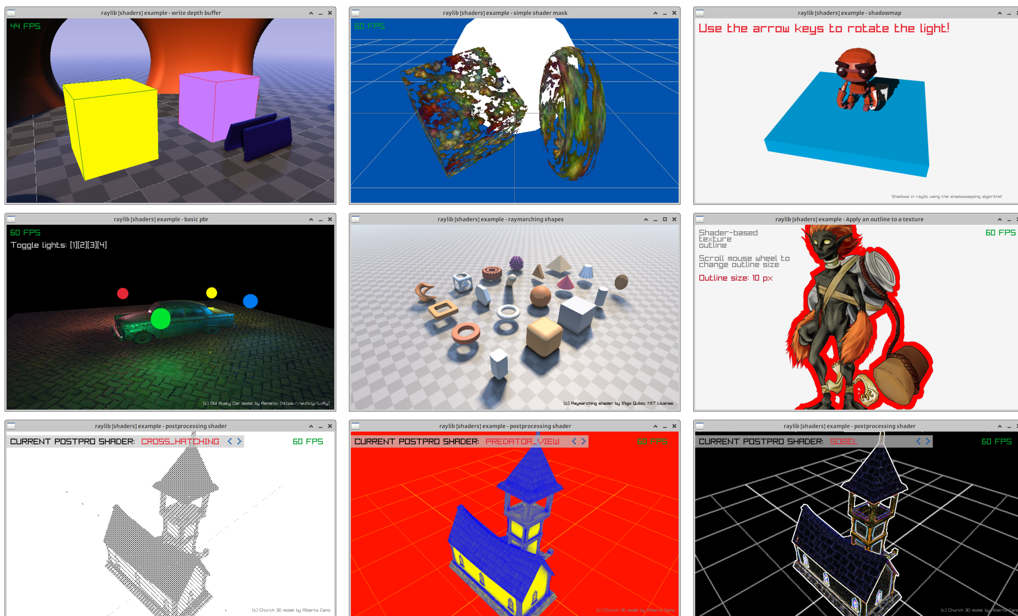
Mohu shrnout zkušenosti z novějšího notebooku s Xubuntu a OpenGL v4.3. Instalace Raylibu byla vzorná dle návodu, předvolená je verze 3.2 OpenGL, nebylo tedy potřeba zasahovat. Po přidání zmíněného řádku do ~/`.bashrc` mi šla i instalace RaylibR přímo v R. Poznámka: v pozadí RaylibR bere Raylib verze 4.0, je však již verze 5.0. Je stále co zlepšovat!

Zde je drobný dávkový soubor, který vám spustí všechny zkompilované ukázky, jednu za druhou. Napočítal jsem jich 146+1 šablona. Soubor mám uložený ve složce `raylib-master/`.

```
# Prázdný řetězec pro všechny, či vybranou složku z:
# audio core models others shaders shapes text textures
malkde=""
cd examples/$malkde
for soubor in `find . -type f -executable | sort`; do
echo "Spouštím $soubor..."
./$soubor # tichý režim: ./$soubor 1>/dev/null 2>/dev/null
done
```

Ukázky jsou výtečné, minimálně animovaná `raylibr::julia` v R či ukázka `shaders_julia_set` v Raylibu je neokoukaná (obrázky výš, vpravo).

Pro potěšení oka, z těch dalších, které mi v Raylib v5.0 na staříčkovi nejely, zmíním ze složky `examples/shaders/`, prefix `shaders_`, tyto ukázky: `write_depth`, `simple_mask`, `shadowmap` na prvním řádku, `raymarching`, `basic_pbr`, `texture_outline` na 2. řádku a na 3. řádku z `postprocessing`, speciálně cross_hatching, predator_view a sobel.



8. Symbolická poznámka

Název článku má být „pomsta“ za ten boj s instalacemi. Při $x + y = x \cdot y$, tedy $y = x/(x-1)$ či $x = y/(y-1)$, ani jeden z programů nemůže být sám o sobě jedničkou. Raylib je skvělý na hry, ale o světě R neví nic. Naopak R má na vizualizaci tohoto typu ještě dost velké rezervy. Ale oba programy mohou být zároveň nula, to byl ten případ, kdy se nedařilo nainstalovat Raylib ani RaylibR, ze začátku to nešlo a nešlo…

Držím palce, ať je vždy $x + y > 1$! Třeba dvě dvojky, nemusí to být hned jedničky s hvězdičkou.

Kontaktní adresa

Ing. Pavel Stříž, Ph.D., U Škol 940, Bučovice, okres Vyškov, 685 01, Česká republika,
E-mailová adresa: `pavel@striz.cz`